

# Dictionary-based Tensor Canonical Polyadic Decomposition

Jérémy Emile Cohen\* and Nicolas Gillis

## Abstract

To ensure interpretability of extracted sources in tensor decomposition, we introduce in this paper a dictionary-based tensor canonical polyadic decomposition which enforces one factor to belong exactly to a known dictionary. A new formulation of sparse coding is proposed which enables high dimensional tensors dictionary-based canonical polyadic decomposition. The benefits of using a dictionary in tensor decomposition models are explored both in terms of parameter identifiability and estimation accuracy. This is illustrated on the decomposition of simulated data and on the unmixing of hyperspectral images.

## Index Terms

tensor, multiway analysis, sparse coding, constrained optimization, spectral unmixing.

## I. INTRODUCTION

Given a mixture of several components, a classic problem in signal processing is to separate the contribution of each of these components using solely the information contained in the data. This problem is known as blind source separation and has been a particularly widely studied topic for the last two decades [1]. When the available data are contained in a multiway array, that is, a table of three entries or more, blind source separation has been successfully achieved by means of tensor decomposition techniques in a large variety of applications, ranging from telecommunications [2] to chemometrics [3], spectral unmixing [4], neuroimaging [5], social sciences [6] and machine learning [7].

The key concept behind tensor decomposition methods is the linearity of blocks of parameters of interest such as spectra, concentrations or time signatures with respect to experimental parameters such as wavelength, pixel index, subject index or time. For the canonical polyadic decomposition (CPD) model studied in this paper, each block of parameters of interest depends respectively on only one experimental parameter.

As an illustration, tensor decomposition techniques can be used to perform source separation with hyperspectral images, a task often referred to as spectral unmixing. Hyperspectral images are 2D images collected for a large number of wavelengths, and possibly along time. In this scenario, the blocks of parameters of interest are spectral signatures (characteristic responses of materials to light stimulation), abundances (relative concentrations) and time evolution.

For most applications, tensor decomposition models are not exactly matching the physical models because of their strong assumptions. As a consequence, modeling error has to be accounted for when processing the results of the source separation using tensors. This matters when the goal of the source separation is to identify the components in the mixture. The estimated parameters of interest have to be compared with benchmarks, and modeling error induces error on the estimated parameters which may in turn deteriorate identification performances.

In this paper, we want to avoid splitting the identification procedure and the source separation procedure. It is shown that using a formalism inspired from sparse coding [8], merging source separation and identification is not only possible but also offers advantages in term of uniqueness properties of the tensor

The authors acknowledge the support by the F.R.S.-FNRS (incentive grant for scientific research n° F.4501.16). NG also acknowledges the support by the ERC (starting grant n° 679515).

University of Mons 7000 Mons, Belgium (e-mail: firstname.lastname@umons.ac.be).

Manuscript published on arxiv beginning of 04/2017.

decomposition, and may reduce estimation error on identified factors provided the *a priori* information is accurate.

### OUTLINE AND CONTRIBUTIONS

In this work, after discussing notations and vocabulary, we make the following contributions:

- Formalize high order tensor sparse coding by modifying the usual sparse coding formulation for matrices [8], [9]. We call the obtained tensor model dictionary canonical polyadic decomposition (DCPD).
- Provide tools for introducing flexibility in the DCPD formulation. This makes the DCPD model more suitable for practical problems.
- Study the identifiability of parameters of the DCPD for matrices and higher order tensors, and study the existence of a best low rank DCPD approximation.
- Develop greedy and continuous algorithms to compute the DCPD and its flexible variants.
- Check the identification performances of the DCPD model on synthetic data with respect to the CPD model. We also use DCPD in the matrix case to perform spectral unmixing under pure-pixel assumption of the Urban, San Diego and Terrain data sets<sup>1</sup> and compare our results with state-of-the-art methods [10]–[13].

### NOTATION AND VOCABULARY

Among various notation habits in the multiway array processing community, we choose to follow notation from [14], [15], as presented in Tables I and II. Although our results will be applicable to tensors of any order, we focus in this paper on third order tensor in order to simplify the presentation. We call a **third-order** real  $K \times L \times M$  **tensor**  $\mathcal{T}$  a vector from a tensor space  $(\mathbb{R}^K \otimes \mathbb{R}^L \otimes \mathbb{R}^M, \otimes)$  with  $\otimes$  being a multilinear mapping. A **three way array** is an element from  $(\mathbb{R}^{K \times L \times M}, \otimes)$  where  $\otimes$  is the outer product. In other words, the set of arrays is a tensor space when the outer product is used as the tensor product. A higher order tensor is a vector from a tensor space featuring at least three linear subspaces.

In Table II, some useful properties of multilinear operators acting on tensors are specified. Multilinear operators generalize linear operators acting on vectors by defining linear operations on each vector space  $\mathbb{R}^N$  composing the tensor space. These multilinear operators also form a tensor space  $(\mathbb{R}^{R_1 \times K} \otimes_{op} \mathbb{R}^{R_2 \times K} \otimes_{op} \mathbb{R}^{R_3 \times K}, \otimes_{op})$  where  $\otimes_{op}$  will be abusively denoted  $\otimes$  although it is not the outer product, and  $\otimes_{op}$  is defined by  $(U \otimes_{op} V)(a \otimes b) = Ua \otimes Vb$ .

Given a  $K \times L \times M$  tensor  $\mathcal{T}$ , its **canonical polyadic decomposition** (CPD) of rank  $R$  can be written as follows:

$$\mathcal{T} = \sum_{r=1}^R \mathcal{D}_r, \quad (1)$$

where  $\mathcal{D}_r$  are decomposable tensors of the form  $\mathcal{D}_r = \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r$ . The **rank** of  $\mathcal{T}$  is the minimal value of  $R$  such that (1) holds exactly, while we define the rank of a CPD model as the number of component  $R$  in that model. A CPD model or a tensor is said to be a **low-rank model** or a low-rank tensor if  $R$  is small with respect to all dimensions of the data or tensor.

Finding the CPD of a third-order tensor means finding rank-one tensors  $\mathcal{D}_r$ . Yet, each tensor  $\mathcal{D}_r$  may be defined by three vectors  $\mathbf{a}_r$ ,  $\mathbf{b}_r$  and  $\mathbf{c}_r$ , only up to two scaling ambiguities; in fact,  $\mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r = \alpha \mathbf{a}_r \otimes \beta \mathbf{b}_r \otimes \mathbf{c}_r / \alpha \beta$ ,  $\forall \alpha, \beta \neq 0$ .

Next, it is often convenient to store these vectors in matrices called **factors** as  $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_R]$ ,  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_R]$  and  $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_R]$ . This leads to a convenient writing:

$$\mathcal{T} = (\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C}) \mathcal{I}_R, \quad (2)$$

<sup>1</sup>available at <http://www.agc.army.mil/>

where  $\mathbf{A} \in \mathbb{R}^{K \times R}$ ,  $\mathbf{B} \in \mathbb{R}^{L \times R}$  and  $\mathbf{C} \in \mathbb{R}^{M \times R}$  are called factor matrices, and  $\mathcal{I}_R = \sum_{r=1}^R \mathbf{e}_i \otimes \mathbf{e}_i \otimes \mathbf{e}_i$  belongs to  $\mathbb{R}^{R \times R \times R}$  with  $\mathbf{e}_i$  a canonical basis vector of  $\mathbb{R}^R$ , that is,  $\mathcal{I}_R$  is a diagonal core tensor with only ones on the diagonal. A convenient interpretation of (2) is to see it as a change of basis. Indeed, (2) means that the vector  $\mathcal{T}$  is expressed by coefficients  $\mathcal{I}_R$  in the image of multilinear operator  $\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C}$ , with linear operators  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  spanning respectively the first, second and third **mode** of  $\mathcal{T}$ . Note that model (2) now contains  $2R$  scaling indeterminacies whereas definition (1) did not contain any.

Conditions on the dimensions of the tensor and rank of the decomposition are given in the literature [16]–[18] to ensure uniqueness of the factors in an unconstrained CP model, but only when noise is absent. When these conditions are satisfied, the factors in the CPD model are unique up to permutation ambiguity and scaling described above. The model is then said to be **identifiable**. For higher order tensors, that is, arrays of three ways and more, these conditions are mild.

$\mathcal{E} \otimes \mathcal{F}$ :	tensor product space, linear space mapped by $\otimes$ from $\mathcal{E} \times \mathcal{F}$ .
$\mathbf{a} \otimes \mathbf{b}$ :	tensor product of two vectors, <i>i.e.</i> an element of $\mathcal{E} \otimes \mathcal{F}$ , can be understood as an outer product of vectors if the tensor space is an array space. [14]
$\mathbf{a} \boxtimes \mathbf{b}$ :	Kronecker product between vectors [19].
$\mathbf{A} \odot \mathbf{B}$ :	Khatri-Rao (columnwise Kronecker) product of matrices [20].
$\mathbf{A} \boxtimes \mathbf{B}$ :	Hadamard product of matrices, <i>i.e.</i> , element-wise product [20].

TABLE I  
BASIC DEFINITIONS FROM LINEAR ALGEBRA

Matrices are a particular case of tensors with only two modes. One of the main differences of the CPD for matrices compared with the CPD of higher order tensors is that it is never identifiable without additional constraints as soon as  $R > 1$ . Note that, since the two modes of a matrix are the column space and the row space, the CPD of a matrix can be written as  $\mathbf{M} = \mathbf{A}\mathbf{B}^T$ .

A tool that needs to be introduced here is the unfolding of three-way arrays. By choosing three particular ways to cut the cube of data into slices and stacking the obtained matrices, it is possible to rewrite the CPD in a matrix format. We chose the unfolding defined in [15], yielding the following matrix format CPD :

$$\begin{aligned} \mathbf{T}_1 &= \mathbf{A}(\mathbf{B} \odot \mathbf{C})^T, \\ \mathbf{T}_2 &= \mathbf{B}(\mathbf{A} \odot \mathbf{C})^T, \\ \mathbf{T}_3 &= \mathbf{C}(\mathbf{A} \odot \mathbf{B})^T. \end{aligned} \tag{3}$$

Unfolding  $\mathbf{T}_i$  contains all the vectors along the mode  $i$ . For instance,  $\mathbf{T}_1$  contains all the  $L \times M$  columns of  $\mathcal{T}$ , thus its column space is the subspace spanned on the first mode by  $\mathcal{T}$ , that is, the span of columns of  $\mathbf{A}$ .

$\mathbf{U} \otimes \mathbf{V} \otimes \mathbf{W}$	: an operator acting on a third order tensor.
$(\mathbf{U} \otimes \mathbf{V} \otimes \mathbf{W}) \mathcal{T}$	: application of $\mathbf{U}$ on the first mode, $\mathbf{V}$ on the second mode, and $\mathbf{W}$ on the third mode, also noted $\mathcal{T} \bullet_1 \mathbf{U} \bullet_2 \mathbf{V} \bullet_3 \mathbf{W}$ .

TABLE II  
SOME DEFINITIONS AND PROPERTIES OF MULTILINEAR OPERATORS

## II. DICTIONNARY CPD : MODEL DEFINITION

An important property of low-rank models is that the computed factors bear physical meaning. Factors are interpretable when the multilinear relationship between the experimental parameters and the block of parameters of interest stems from a meaningful modeling, and when the model is identifiable.

The first condition is met in many applications of low-rank models, for instance in fluorescence spectroscopy where spectra and concentrations depend linearly, as a first-order approximation, on the emission wavelength, the excitation wavelength and the mixture index [21]. When mining a collection of text documents, low-rank factorization techniques can identify characteristic words and documents [22].

The identifiability condition may however not be verified in practice. For matrices, it is indeed well known that identifiability is never achieved without imposing additional constraints for  $R$  greater than one, since

$$\mathbf{M} = \mathbf{A}\mathbf{B}^T = \mathbf{A}\mathbf{P}\mathbf{P}^{-1}\mathbf{B}^T \quad (4)$$

for any invertible matrix  $\mathbf{P}$ . In the higher-order case, conditions on the rank of the model and the dimensions of the data have been reported in the literature with some variations, but are mild enough to ensure identifiability of the CPD in most applications [16]–[18].

If the factors are meaningful, it may be of crucial importance to identify them. In the example of fluorescence spectroscopy, once emission and excitation spectra have been extracted from the data using a CPD model, the final step is to recognize which chemical compound is present in the mixtures by matching the extracted spectra with known spectra of known chemicals. The goal of the models presented below is to merge this identification step with the source separation procedure using the *a priori* information available on the factors.

### A. Dictionary-based CPD

Formally, assume the following relationship between a matrix  $\mathbf{D} \in \mathbb{R}^{L \times d}$  called the (over-complete) dictionary and factor  $\mathbf{B}$ :

$$\mathbf{B} = \mathbf{D}\mathbf{S}, \quad \|\mathbf{S}\|_{col,0} = 1, \quad (5)$$

where  $\mathbf{S} \in \{0, 1\}^{d \times R}$  is a binary matrix,  $\|\mathbf{S}\|_{col,0}$  counts the number of non-zero values in each column of  $\mathbf{S}$ , and  $d$  is much larger than  $R$ . Here  $\mathbf{S}$  has exactly one 1 in each column and is a selection matrix that identifies among the  $d$  atoms of the dictionary the  $R$  present in the basis  $\mathbf{B}$ . The chosen columns of  $\mathbf{D}$  that constitute  $\mathbf{B}$  are given by the row index of all the ones in  $\mathbf{S}$ . A priori, no restrictions on the atoms are assumed although some technical conditions to ensure identifiability in particular settings are discussed in section IV.

Viewing (5) as deterministic *a priori* information available on parameter  $\mathbf{B}$ , a **dictionary CPD model (DCPD)** can be written as follows:

$$\begin{cases} \mathcal{T} = (\mathbf{A} \otimes \mathbf{D}\mathbf{S} \otimes \mathbf{C}) \mathcal{I}_R + \mathcal{E}, \\ \|\mathbf{S}\|_{col,0} = 1, \mathbf{S} \in \{0, 1\}^{d \times R}. \end{cases} \quad (6)$$

In this model, the parameters are  $\mathbf{A}$ ,  $\mathbf{S}$  and  $\mathbf{C}$ . In other words,  $\mathbf{B}$  has been parameterized. The tensor  $\mathcal{E}$  represent the noise and typically each entry follows i.i.d. known distributions such as a zero-mean Gaussian distribution. In that case, it is possible to derive the maximum likelihood estimator of parameters  $\mathbf{A}$ ,  $\mathbf{S}$  and  $\mathbf{C}$ :

$$\begin{aligned} & \underset{\mathbf{A}, \mathbf{S}, \mathbf{C}}{\operatorname{argmin}} \|\mathcal{T} - (\mathbf{A} \otimes \mathbf{D}\mathbf{S} \otimes \mathbf{C}) \mathcal{I}_R\|_F^2 \\ & \text{such that } \|\mathbf{S}\|_{col,0} = 1 \text{ and } \mathbf{S} \in \{0, 1\}^{d \times R}. \end{aligned} \quad (7)$$

Of course, the DCPD does not apply in any situation where the CPD works. It is far from obvious that for a given application, a dictionary containing exactly all the factors on one mode is available. Even if such a dictionary is available, it is likely that some variability has to be accounted for between the atoms and the factors. Below, we try to tackle respectively cases where the relation (5) should not be exact, and cases where no dictionary is available.

### B. Accounting for Variability

In spectral unmixing, two different approaches have been studied in the literature: the fully blind unmixing and the semi-blind approach using already known spectra to compute regression [23]. However in this particular application, it is well known that the same material emits a slightly different spectrum depending on additional parameters not accounted for in a low-rank model. These variations are called spectral variability and are a major issue to using both blind unmixing and semi-blind unmixing [24]. Learning from this example, we emphasize that a naive DCPD as introduced above in (6) may be viewed as unrealistic.

Therefore, it may be necessary to introduce flexibility in the DCPD model. A first way is to generalize the relationship between factor  $\mathbf{B}$  and the dictionary  $\mathbf{D}$ :

$$\mathbf{B} = f(\mathbf{D}\mathbf{S}, \theta), \quad (8)$$

for some function  $f(x, \theta)$  mapping to  $\mathbb{R}^{L \times R}$  and where  $\theta$  is a random variable following some known probability  $p(\theta)$ . A simple instance of (8) is obtained by setting  $f(x, \theta) = x + \theta$  and  $p(\theta)$  is a Gaussian distribution of zero mean and known white covariance  $\sigma_c \mathbf{I}_K \otimes \mathbf{I}_R$ . Then (8) is a noisy version of (5) and a maximum *a posteriori* estimator yields (MAP derivation similar to [25]):

$$\begin{aligned} & \underset{\mathbf{A}, \mathbf{B}, \mathbf{S}, \mathbf{C}}{\operatorname{argmin}} \|\mathcal{T} - (\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C}) \mathcal{I}_R\|_F^2 + \frac{1}{\sigma_c^2} \|\mathbf{B} - \mathbf{D}\mathbf{S}\|_F^2 \\ & \text{such that } \|\mathbf{S}\|_{\text{col},0} = 1 \text{ and } \mathbf{S} \in \{0, 1\}^{d \times R}. \end{aligned} \quad (9)$$

In the case where atoms can be grouped with sufficient group population, the user may want to identify the columns of factor  $\mathbf{B}$  with a group of atoms rather than with a single element of  $\mathbf{D}$ . This is the method used in [26] for hyperspectral imaging, where the dictionary contains multiple spectra for each class of mineral. In this scenario, a solution is to cluster the available atoms and use the centroids as columns of  $\mathbf{D}$  along with inter-class covariance. Then comparing  $\mathbf{B}$  to multiple class of atoms with known averages and covariances amounts to penalizing using the Mahalanobis distance [26] in (9).

There are other ways to account for discrepancy between the dictionary and the true phenomena underlying the data. In particular, it is not unreasonable that this discrepancy depends on the other experimental parameters (that is, the first or third mode in our case). For instance, in hyperspectral imaging, a modification of the geometry of the ground due to weather conditions can modify the spectra. If the data is collected along time, then the spectra of a particular material can relate to one atom in the dictionary but with a time-dependent discrepancy. Since modeling variability is a topic-dependent task, we only provide one instance of a modified DCPD that handles this task, to serve as a guide for further works.

*Dictionary-based PARAFAC2:* A well-known modified CPD model that accounts for variability is the PARAFAC2 model [27], [28]. Here we show that the PARAFAC2 model can be adapted incorporating dictionary information. The model is the following: for all positive integer  $m$  smaller than  $M$ ,

$$\begin{cases} \mathbf{M}_m = \mathbf{A} \operatorname{Diag}(\mathbf{c}_m) \mathbf{B}_m^T, \\ \mathbf{B}_m = \mathbf{P}_m \mathbf{B}^*, \\ \mathbf{P}_m^T \mathbf{P}_m = \mathbf{I}_R, \\ \mathbf{B}^* = \mathbf{D}\mathbf{S}, \|\mathbf{S}\|_{\text{col},0} = 1, \mathbf{S} \in \{0, 1\}^{d \times R}, \end{cases} \quad (10)$$

where  $\mathbf{M}_m \in \mathbb{R}^{K \times L}$  is the  $m^{\text{th}}$  slice of data tensor  $\mathcal{T}$  along the third mode,  $\mathbf{c}_m$  is the  $m^{\text{th}}$  row of a matrix of parameters  $\mathbf{C}$ , matrices  $\mathbf{P}_m$  with orthogonal columns are unknown and  $\mathbf{B}^*$  is an unknown latent factor related to all the factors  $\mathbf{B}_m$ . Dictionary  $\mathbf{D}$  has sizes  $D \times d$  where  $D$  can be different from  $M$ .

Here the three-way data is seen as a collection of matrices with shared first mode factor  $\mathbf{A}$  and similar factors  $\mathbf{B}_m = \mathbf{P}_m \mathbf{B}^*$ . The matrix  $\mathbf{B}^*$  stands for a latent shared factor among matrices  $\mathbf{B}_m$ . To include additional knowledge provided by the dictionary, it is assumed that this  $\mathbf{B}^*$  follows equation (5). Because the columns of  $\mathbf{P}$  are orthogonal, the underlying hypothesis here is that all  $\mathbf{B}_m$  have the same covariance

matrix, which is a relaxed assumption with respect to the underlying hypothesis in (6), that is,  $B_m = B^*$  for all  $m$ .

An hidden advantage of the PARAFAC2 version of DCPD is that row dimension of the dictionary can be different from the number of samples on the second mode, since  $P_m$  does not need to be square. In practice this means that when using the PARAFAC2 dictionary model, the measured spectral bands can be different than the spectral bands at which atoms are sampled.

### C. Self-Dictionary

If no dictionary is available on any of the factor, it is still possible in some specific cases to obtain the dictionary from the data. The topic of learning a dictionary from auxiliary data will be dealt with in future works. Rather, we focus here on the **separability** assumption [29], [30], that is when the data itself can be used as a dictionary. In the matrix case, the separability has been well studied in the context of nonnegative matrix factorization (NMF), see next section for details. With the suggested formalism, the self-dictionary CPD (SDCPD) for matrices is written as follows:

$$M = A (M^T S)^T, \quad \|S\|_{col,0} = 1, \quad S \in \{0, 1\}^{d \times R}. \quad (11)$$

A working hypothesis of this model for matrices is that the basis is present in the data, that is, the second mode factor is a subset of the rows of  $M$  which is the dictionary.

A generalization of the separability assumption to higher order tensors is however not straightforward. Recently, an attempt was made to define “pure slices” [31], but another possible generalization is obtained by supposing the columns of factor  $B$  are contained in all the  $K \times M$  columns of the unfolding matrix  $T_2$ . The drawback of this model is the possibly very large number of correlated atoms in the obtained dictionary  $T_2$ . There are various approaches to extend the separability assumption to high order tensors, but in such limited space this paper will only deal with the case where  $D = T_2$ .

## III. RELATED WORKS

### A. Sparse coding

Dictionary CPD presented in this paper can be seen as an extension of sparse coding for higher order tensors, see for instance [8], [9]. In sparse coding, a dictionary  $D$  is available, and the data is to be expressed as linear combinations of a small number of  $R$  atoms:

$$M = DX, \quad \|X\|_{row,0} \leq R. \quad (12)$$

Assuming Gaussianity of the error, the objective function  $\|M - DX\|_F^2$  is convex with respect to matrix  $X$ . This model is equivalent to dictionary CPD for matrices by setting  $SB^T = X$ . Nevertheless there is a crucial differences between the two formulations:

- By splitting the variable  $X$  into two variables  $SB^T$ , the linear constraint  $M = DX$  becomes non-convex with respect to  $S$  or  $B$ .
- The sparse coding formulation involves a large number of variables,  $d \times m$  if  $M$  is  $n$  by  $m$ , whereas the DCPD formulation for matrices involves  $(d + m) \times R$  parameters and  $d \times R$  of these parameters are binary.

In summary, sparse coding involves much more parameters but yields a convex cost function.

Since the mixed pseudo norm  $\ell_{row,0}$  is non-convex, most methods for computing sparse coding rely on convex relaxations. Row-sparsity of  $X$  can be achieved in different ways; in particular using convexifications based on the  $\ell_1$  norm, e.g.,  $\ell_{1,2}$  [9], [32],  $\ell_{1,\infty}$  [33], or using linear programming [34], [35]. As far as we know, it is the first time sparse coding will be tackled using the reformulation  $X = SB^T$ . Although the resulting problem will be non-convex, it will involve significantly less variables hence will be applicable to large-scale problems. We will present in Section V several algorithms to tackle this new formulation.

Recently, Salhoun et al. suggested a direct extension of sparse coding for higher order tensors [36] in the particular context of harmonic retrievals, but an obvious difficulty is that the row-sparsity has to be imposed on Khatri-Rao products of factors.

### B. NMF with self dictionary in spectral unmixing

Similarly, self dictionaries have been studied for matrices using the sparse coding formalism, by setting  $\mathbf{D} = \mathbf{M}$ . This is often called the pure-pixel assumption in spectral unmixing [23], since the self-dictionary model assumes some columns of the matrix  $\mathbf{M}$  are not mixtures of more than one column of  $\mathbf{B}$ . As far as we know, there are mainly two types of approaches to tackle (11):

- *Geometric* approaches that selects the atoms in the dictionary based on some geometric criteria, typically based on the volume of the convex hull of  $\mathbf{M}^T \mathbf{S}$ . These approaches include for example vertex component analysis (VCA) [11] and the successive projection algorithm (SPA) [13], [37]–[39]. They are usually fast, running in  $\mathcal{O}(mnR)$  operations. However, they do not always select atoms leading to a small data fitting term  $\|\mathbf{M} - \mathbf{A}(\mathbf{M}^T \mathbf{S})^T\|_F$  since, most of them do not take it into account directly, as they usually put an emphasis on some geometric properties of  $\mathbf{M}^T \mathbf{S}$  (such as having a large volume). In particular, these methods are in general sensitive to outliers.
- *Sparse regression* approaches that are based on the sparse coding reformulation of (11)

$$\begin{aligned} \min_{\mathbf{X} \in \mathbb{R}_+^{d \times n}} \|\mathbf{M} - \mathbf{X}\mathbf{M}\|_F^2 \\ \text{such that } \mathbf{X} \text{ has } r \text{ non-zero columns,} \end{aligned}$$

and achieve row sparsity constraints on the scores  $\mathbf{M}$  in different ways [9], [32]–[35]. These methods have the advantage to better model (11) than geometric approaches as they take into account the data fitting term explicitly. They usually provide good solutions but are rather costly as an optimization problem in  $dn$  variables must be solved, where  $n$  stands for the column dimension of  $\mathbf{M}$ . In particular, for  $\mathbf{D} = \mathbf{M}^T$ , we have  $d = n$  hence  $n^2$  variables. In hyperspectral unmixing,  $n$  is the order of millions and these approaches are impractical. Hence pixels have to be selected in a preprocessing step [40] (e.g., using a geometric approach). Moreover, the problem solved is an approximation of the original problem, which results may not be as close as desired to the solutions of the non-convex problem.

In section V, we describe several algorithms to tackle the proposed formulation (11). They combine the advantages of the two types of approaches described above: they are fast, running in  $\mathcal{O}(mnR)$  operations, but taking explicitly the data fitting term  $\|\mathbf{M} - \mathbf{X}\mathbf{M}\|_F^2$  into account.

### C. Constrained tensor decompositions

The DCPD can be understood as a constrained CPD model. It is similar in spirit to computing CPD when a basis of representation is known for one of the factors, which typically happens when using Tucker Decomposition [41] or a basis of splines [42] for compression. The main difference is that bases of representations are not over-complete, thus no sparsity constraints are imposed on the coefficients.

Using a dictionary to help with recovering factor  $\mathbf{B}$  of the CPD is closely related to parameterizing the columns of that factor. Parametrization is a viable option when an analytical formulation of the atoms of the dictionary is possible; see for instance [43, section IV] and references therein. This may yet not always be the case in real-life applications. The motivation behind the two methods is however the same: reduce the number of degrees of freedom in the tensor decomposition model by providing a set of admissible solutions to improve estimation accuracy, and to restore identifiability in some pathological cases, some of which are discussed in the next section.

#### D. Abundances estimation in hyperspectral unmixing

In the context of spectral unmixing of hyperspectral images, using a known library of spectra to estimate the second factor  $\mathbf{B}$  is a widely studied topic. The matrix  $\mathbf{A}$  contains the abundances and refers to relative concentrations of materials on each pixel. Except for the methods described above, the most widely used techniques to compute  $\mathbf{A}$  when  $\mathbf{D}$  is known are, to the best of our knowledge, MESMA [44], MELSUM [45], BSMA [46] and AutoMCU [47], which all rely on a more or less exhaustive search of all combinations of atoms in  $\mathbf{D}$ , except AutoMCU which only draws randomly a subset of possible atoms. For all possible combinations, abundances are computed, and the best abundances are those that minimize reconstruction error in addition to satisfying some interpretability criteria.

The proposed approach differs from these techniques since it merges blind source separation techniques for spectral unmixing, namely NMF, but featuring atom selection. To the best of our knowledge, our approach has not been described yet in the spectral unmixing literature, and is bound to be computationally less expensive than an exhaustive search.

### IV. IDENTIFIABILITY

The following section contains partial results on the identifiability of the DCPD model, cast in particular cases of interest.

#### A. Matrix case

It is well known that for matrices, the low rank CPD model is not identifiable because of the rotation ambiguity, see equation (4). However, when a dictionary is available for one of the modes, this rotation ambiguity may be fixed given some conditions on  $\mathbf{D}$ :

**Proposition 1.** *Let  $\mathbf{M}$  be a real  $n \times m$  matrix, and let  $\mathbf{D}$  be a real  $n \times d$  matrix with  $\text{spark}(\mathbf{D}) \geq 2R$  where  $\text{spark}(\mathbf{D})$  is the minimum integer  $k$  such that any subsets of  $k$  columns of  $\mathbf{D}$  is full rank. If there exists  $\mathbf{S} \in \{0, 1\}^{d \times R}$  with  $\|\mathbf{S}\|_{\text{col},0} = 1$  and  $\mathbf{A} \in \mathbb{R}^{n \times R}$  with nonzero columns such that  $\mathbf{M} = \mathbf{A}(\mathbf{D}\mathbf{S})^T$ , then  $\mathbf{S}$  and  $\mathbf{A}$  are unique up to permutation ambiguity.*

*In other words, if there exists  $\mathbf{S}' \in \{0, 1\}^{d \times p}$  with  $\|\mathbf{S}'\|_{\text{col},0} = 1$ , and  $\mathbf{A}' \in \mathbb{R}^{n \times p}$  with nonzero columns such that  $2p \leq \text{spark}(\mathbf{D})$  and  $\mathbf{M} = \mathbf{A}'(\mathbf{D}\mathbf{S}')^T$ , then  $p = R$ ,  $\mathbf{S}' = \mathbf{S}$  and  $\mathbf{A}' = \mathbf{A}$ .*

*Proof.* Since  $\text{spark}(\mathbf{D}) > 2R \geq R$ ,  $\mathbf{B} = \mathbf{D}\mathbf{S}$  has full column rank hence  $\mathbf{A}$  is unique in the decomposition  $\mathbf{M} = \mathbf{A}\mathbf{B}^T$  if  $\mathbf{B}$  is unique. Suppose there exists non-equal  $\mathbf{B} = \mathbf{D}\mathbf{S}$  in  $\mathbb{R}^{L \times R}$  and  $\mathbf{B}' = \mathbf{D}\mathbf{S}'$  in  $\mathbb{R}^{L \times p}$  so that  $\mathbf{M} = \mathbf{A}\mathbf{B} = \mathbf{A}'\mathbf{B}'$  for some matrices  $\mathbf{A}, \mathbf{A}'$  with nonzero columns. Because  $\mathbf{B}$  and  $\mathbf{B}'$  both span the row space of  $\mathbf{M}$ , there exist  $R + p$  linearly dependent columns of  $\mathbf{D}$ . By contraposition, if  $\text{spark}(\mathbf{D}) \geq R + p$ ,  $\mathbf{B} = \Pi\mathbf{B}'$  for some permutation operator  $\Pi$ .  $\square$

Proposition 1 shows that if the set of non-trivial DCPD of a matrix for a decomposition rank smaller than the spark of the dictionary is non-empty, then it is a singleton. Interestingly, the constraint on the spark of the dictionary can be relaxed if  $R$  is set to the rank of matrix  $\mathbf{M}$ . Then  $\mathbf{B}$  spans exactly the row space of  $\mathbf{M}$  and the sufficient condition becomes  $\text{spark}(\mathbf{D}) \geq R + 1$ .

#### B. CPD and DCPD uniqueness

For all applications where the identifiability of the CPD is usually verified, a natural question to ask is whether the DCPD will automatically be identifiable or if some additional conditions need to be checked before trying to use the DCPD model. It turns out that given the uniqueness of the CPD up to scaling and permutations of the factors, the only requirement to obtain uniqueness of the DPCD is the uniqueness of the factorization  $\mathbf{B} = \mathbf{D}\mathbf{S}$ , which itself is quite simple to check.

**Proposition 2.** *Let  $\mathcal{T} \in \mathbb{R}^{K \times L \times M}$  be such that  $\mathcal{T}$  admits a unique rank  $R$  CPD up to scaling and permutations. Then if  $\mathbf{D}$  does not contain colinear atoms and there exist  $\mathbf{S}$  verifying (5), the DCPD is also unique up to scaling and permutations.*



*Proof.* Because the CPD of  $\mathcal{T} = (\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C}) \mathcal{I}$  is unique, the DCPD is unique if and only if  $\Omega = \{\mathbf{S} | \mathbf{B} = \mathbf{D}\mathbf{S}\}$  is a singleton. The set  $\Omega$  is not empty by assumption, and if  $\mathbf{S}_1, \mathbf{S}_2$  belong to  $\Omega$ , then the columns they select in  $\mathbf{D}$  have to be colinear (or equal if there was no scaling ambiguity on  $\mathbf{B}$ ). By hypothesis this implies  $\mathbf{S}_1 = \Pi \mathbf{S}_2$ , where  $\Pi$  is a column permutation operator.  $\square$

The existence of  $\mathbf{S}$  is what is really difficult to assess in practice. On the other hand, the dictionary may contain very correlated atoms, but not exactly colinear atoms, so that the DCPD as a model is identifiable in practice whenever the CPD is identifiable.

Moreover, using proposition 1, it is possible to derive a mild sufficient condition for uniqueness of DCPD.

**Proposition 3.** *Let  $\mathcal{T}$  be a real  $K \times L \times M$  tensor of rank  $R$  and  $\mathbf{D}$  a real  $L \times d$  matrix. If there exists  $\mathbf{S} \in \{0, 1\}^{d \times R}$  with column sparsity set to 1, and  $\mathbf{A} \in \mathbb{R}^{K \times R}$  and  $\mathbf{C} \in \mathbb{R}^{M \times R}$  such that  $\mathcal{T} = (\mathbf{A} \otimes \mathbf{D}\mathbf{S} \otimes \mathbf{C}) \mathcal{I}_R$ , if  $\text{spark}(\mathbf{D}) \geq 2R$  and if  $\mathbf{A}$  and  $\mathbf{C}$  do not have zero columns, then  $\mathbf{S}$ ,  $\mathbf{A}$  and  $\mathbf{C}$  are unique up to permutation and scaling ambiguity.*

*Proof.* By applying Proposition 1 to  $\mathbf{T}_2 = \mathbf{D}\mathbf{S}(\mathbf{A} \odot \mathbf{C})^T$ ,  $\mathbf{S}$  and the Khatri-Rao product  $\mathbf{P} = \mathbf{A} \odot \mathbf{C}$  are unique up to permutation ambiguity. It remains to prove that the decomposition of  $\mathbf{P}$  into  $\mathbf{A} \odot \mathbf{C}$  is unique provided that the dimensions of the problem are fixed. Column-wise, we need to check that  $\mathbf{p}_i = \mathbf{a}_i \boxtimes \mathbf{c}_i$  is unique. Because the dimensions of the tensor are fixed, there exist a unique matricization operator so that  $\text{matr}(\mathbf{p}_i) = \mathbf{a}_i \mathbf{c}_i^T$ . This rank-one approximation is indeed unique up to scaling, and because the matricization operator is an isomorphism, the decomposition of  $\mathbf{P}$  is therefore unique.  $\square$

An important remark is that Proposition 3 is true for any order, since rank-one decompositions are always unique up to scaling. Similar discussion on identifiability when one factor is unique can be found in [48] and references therein. Moreover, if  $\mathbf{A}$  has many colinear columns as may be the case with spectra in spectral unmixing, if no factor has a zero column, then the DCPD model does not suffer from the rotation ambiguity inherent to the CPD with colinear columns in factors. Colinear columns in factors appear when multiple columns of a factor in one mode are necessary to express the evolution of only one physically meaningful component.

### C. Existence of the best low-rank DCPD

For third-order tensors, another important advantage of DCPD is that it makes the optimization problem well-posed. In fact, for CPD, the optimal solution may not exist as the feasible set is open; see for example [49].

**Proposition 4.** *Let  $\text{spark}(\mathbf{D}) > R$ , then the optimal solution of (7) is attained.*

Below we provide two different proofs, each shedding a different light on the existence of the best low rank DCPD approximation.

*First proof.* Let us show that the set  $\mathcal{E}_B = \{\mathbf{B}(\mathbf{A} \odot \mathbf{C}) \mid \mathbf{A} \in \mathbb{R}^{K \times R}, \mathbf{C} \in \mathbb{R}^{M \times R}\}$  is closed. First, notice that  $\mathcal{E}_B$  is the image of  $\mathcal{E} = \{\mathbf{A} \odot \mathbf{C} \mid \mathbf{A} \in \mathbb{R}^{K \times R}, \mathbf{C} \in \mathbb{R}^{M \times R}\}$  by a full column rank linear operator. Indeed,  $\text{spark}(\mathbf{D}) > R$ . Then it is sufficient to show that  $\mathcal{E}$  is closed, since  $\mathcal{E}_B = (\mathbf{B}^\dagger)^{-1}(\mathcal{E}) \cap \text{col}(\mathbf{B})$ , where  $^\dagger$  is the left pseudo-inverse and  $\text{col}(\mathbf{B})$  is the column space of  $\mathbf{B}$ . Finally, since the set of rank-one matrices is closed, and the matrices in  $\mathcal{E}$  are columnwise vectorized rank-one matrices,  $\mathcal{E}$  is closed.  $\square$

*Second proof.* Since  $\text{spark}(\mathbf{D}) > R$ , for any of  $\mathbf{S}$ ,  $\mathbf{B} = \mathbf{D}\mathbf{S}$  has rank  $R$ . Note that there are a finite number of such  $\mathbf{S}$ . It remains to show that for any  $\mathbf{B}$  of rank  $R$ , the infimum of

$$\inf_{\mathbf{A}, \mathbf{C}} \|\mathcal{T} - (\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C}) \mathcal{I}_R\|_F^2$$

is attained. Without loss of generality, we can assume that  $\|\mathbf{a}_r\|_2 = 1$  for  $1 \leq r \leq R$  by the scaling degree of freedom of each rank-one tensor  $\mathcal{D}_r = \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r$ . Using unfolding, the infimum of the above problem is attained if and only if the infimum of

$$\inf_{\mathbf{A}, \|\mathbf{a}_r\|_2=1 \forall r, \mathbf{C}} \|\mathbf{T}_3 - \mathbf{C}(\mathbf{A} \odot \mathbf{B})^T\|_F^2$$

is attained. Moreover, since  $\mathbf{C} = 0$  is a feasible solution, we can add the constraint

$$\|\mathbf{T}_3 - \mathbf{C}(\mathbf{A} \odot \mathbf{B})^T\|_F \leq \|\mathbf{T}_3\|_F$$

implying  $\|\mathbf{C}(\mathbf{A} \odot \mathbf{B})^T\|_F \leq 2\|\mathbf{T}_3\|_F$ . For  $\text{rank}(\mathbf{B}) = R$  and  $\|\mathbf{a}_r\|_2 = 1 \forall r$ , we can show that

$$\sigma_R(\mathbf{A} \odot \mathbf{B}) \geq \sigma_R(\mathbf{B}) > 0,$$

where  $\sigma_R(\mathbf{B})$  is the  $R$ th singular value of  $\mathbf{B}$ . In fact, denoting  $\mathbf{a}^j$  the  $j$ th row of  $\mathbf{A}$ ,

$$\begin{aligned} \sigma_R^2(\mathbf{A} \odot \mathbf{B}) &= \min_{\|x\|_2=1} \|(\mathbf{A} \odot \mathbf{B})x\|_2^2 \\ &= \min_{\|x\|_2=1} \sum_j \|\mathbf{B}(x \boxtimes \mathbf{a}^j)\|_2^2 \\ &\geq \sigma_R^2(\mathbf{B}) \min_{\|x\|_2=1} \sum_j \|x \boxtimes \mathbf{a}^j\|_2^2 \\ &= \sigma_R^2(\mathbf{B}) \min_{\|x\|_2=1} \sum_{r=1}^R |x_r| \|\mathbf{a}_r\|_2^2 = \sigma_R^2(\mathbf{B}). \end{aligned}$$

Finally,  $\|\mathbf{C}\|_F \leq 2 \frac{\|\mathbf{T}_3\|_F}{\sigma_{\min}(\mathbf{B})}$  hence the feasible set can be reduced to a compact set hence the infimum is attained since the objective function is continuous and bounded below.  $\square$

## V. DECOMPOSITION ALGORITHMS : GREEDY AND NON-GREEDY APPROACHES

In the literature of sparse approximation, two families of algorithms have been studied extensively: greedy approaches based on matching pursuit, and continuous approaches based on convex relaxations of the  $\ell_0$  pseudo norm [50], [51]. In the same spirit, we develop in the next two sections the two same kind of algorithms to attack (6). We will compare these approaches in Section VI.

This paper does not explicitly discuss algorithms for computing the CPD itself. In DCPD, factor matrices  $\mathbf{A}$  and  $\mathbf{C}$  can be estimated using any off-the-shelf CPD algorithm assuming  $\mathbf{B}$  is fixed. This can be done for instance using the alternating least squares procedure in the unconstrained case, or exact non-negative least squares for non-negative CPD [52], [53]. Therefore, below, only the estimation of  $\mathbf{S}$  and  $\mathbf{B}$  for fixed  $\mathbf{A}$  and  $\mathbf{C}$  is discussed<sup>2</sup>.

### A. Greedy algorithms : matching pursuit

Let us first provide an algorithm to compute the minimum (7) with respect to  $\mathbf{S}$ . Since the set of solution for  $\mathbf{S}$  is discrete, the underlying optimization problem is combinatorial. On the other hand, computing the unconstrained CPD can be done efficiently using alternating least squares (ALS). We want to take advantage of both unconstrained CPD and greedy algorithms for computation efficiency when estimating respectively factors  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  and the selection matrix  $\mathbf{S}$ .

This is what the matching pursuit-ALS (MPALS) described below does. The variable  $\mathbf{B}$  is injected in the problem to be a proxy of  $\mathbf{D}\mathbf{S}$ . The matrix  $\mathbf{B}$  is estimated using the least square update<sup>3</sup>

$$\hat{\mathbf{B}} = \mathbf{T}_2(\mathbf{A} \odot \mathbf{C})^\dagger = \mathbf{T}_2(\mathbf{A} \odot \mathbf{C})(\mathbf{A}^T \mathbf{A} \boxtimes \mathbf{C}^T \mathbf{C})^{-1}, \quad (13)$$

<sup>2</sup>Algorithms introduced in this section are available at <https://jeremy-e-cohen.jimdo.com/downloads/>

<sup>3</sup>The inverse is not actually computed, rather we solve the least square problem using any efficient solver. The bottleneck here is the large product  $\mathbf{T}_2(\mathbf{A} \odot \mathbf{C})$  since  $R \ll \min(K, L, M)$ .

and  $\hat{S}$  is then evaluated by choosing the closest atom in  $D$  up to as scaling factor. Finally  $B$  is reevaluated as  $\hat{B} = D\hat{S}$ . In other words,  $B$  is estimated through a projected least squares update, where the projection space is the set spanned by  $DS$  for all  $S$  with coefficients in  $\{0, 1\}$  and column sparsity set to 1.

---

**Algorithm 1** Matching Pursuit Alternating Least Squares

---

**INPUT:** array  $\mathcal{T}$ , factors  $A$  and  $C$ , dictionary  $D$ .

**$B$  estimate:**  $B = \mathcal{T}_{(2)}(A \odot C)(A^T A \boxminus C^T C)^{-1}$

**$S$  estimate:**

**for**  $i$  from 1 to  $R$  **do**

$S_{j^*i} = 1 \iff j^* = \operatorname{argmax}_j \frac{\langle B_i | D_j \rangle}{\|D_j\|}$

**end for**

**$B$  reevaluation:**  $B = DS$

**OUTPUT:** Estimated scores  $S$  and factor  $B$ .

---

Like most projected ALS algorithms such as the ALS algorithm for NMF<sup>4</sup> [54], convergence of the global MPALS algorithm, that is including the alternating least squares estimation of  $A$  and  $C$ , cannot be ensured since the cost function is not guaranteed to decrease at each step. Indeed, the least squares update of  $B$  decreases the cost function, but the projection step increases it. Therefore MPALS is bound to have few provable results in term of convergence, contrary to continuous algorithms presented below. It is however a very simple algorithm to implement with no parameter to tune and it provides good results in both simulated and real data experiments reported in Section VI. In particular, we observed convergence in practice, and the cost function decreases for most iterations in simulations. The complexity of  $\mathcal{O}(RKLM + RLd)$  operations per iteration of MPALS inside an ALS algorithm is about the same as plain ALS if the dictionary is not excessively large.

Note that the term greedy is a bit abusive since an atom chosen to belong to  $B$  at some iteration of the global procedure may be discarded at a further iteration. What is greedy in MP-ALS is the procedure to choose atoms in the dictionary at each inner iteration. Also, MP-ALS can be easily adapted if the constraint on the number of elements in  $S$  is modified to allow for more than one atom to be used to approximate the columns of  $B$ , thus the borrowed name matching pursuit.

MPALS can also easily be adapted to tackle other similar optimization problems. If the factors are constrained to be nonnegative, then the estimates of factors  $A$  and  $C$  can be obtained by nonnegative least squares as mentioned earlier, while factor  $B$  should be nonnegative since the dictionary should be nonnegative in this case.

*A smooth version of MPALS:* MPALS features a projection of  $B$  on the atoms of the dictionary at each outer iteration, which is a very rough way to impose the sparsity constraint on the scores  $S$ . To obtain a smoother optimization algorithm, we suggest to enforce sparsity constraints in a continuous manner using the flexible formulation of the DCPD for factor  $B$ , while using the projected factor  $DS$  when estimating the other factors. We call this algorithm smooth MPALS (SMPALS), it is summarized in Algorithm 2. This algorithm also finds the minimum of (9) solving the flexible dictionary problem with Gaussian noise. MPALS is modified in a straightforward manner, by making the least squares update of  $B$  depend on  $S$ , and not reevaluate  $B$  after evaluating  $S$ . This means removing  $B = DS$  in Algorithm 1, and replacing  $B$  first estimate with

$$\hat{B} = (T_2(A \odot C) + \lambda DS)(A^T A \boxminus C^T C + \lambda I_{R \times R})^{-1} \quad (14)$$

where  $\lambda$  is a given parameter, set by the user at the beginning of the algorithm, meant to approach  $\frac{1}{\sigma_c^2}$  in (9).

In the alternating outer loop,  $B$  can be set either to the exact  $DS$ , or to the approximate version computed by (14). The first choice provides the algorithm we called S-MPALS, while the second is

<sup>4</sup>In Matlab, this is the ‘als’ algorithm of ‘nnmf’.

**Algorithm 2** Smooth Matching Pursuit Alternating Least Squares

---

**INPUT:** array  $\mathcal{T}$ , factors  $\mathbf{A}$  and  $\mathbf{C}$ , dictionary  $\mathbf{D}$ , coupling parameter  $\lambda > 0$  and update rate  $p > 1$ .

 **$\mathbf{B}$  least squares estimate:**

$$\mathbf{B} = (\mathcal{T}_{(2)} (\mathbf{A} \odot \mathbf{C}) + \lambda \mathbf{D} \mathbf{S}) (\mathbf{A}^\top \mathbf{A} \boxplus \mathbf{C}^\top \mathbf{C} + \lambda \mathbf{I}_{R \times R})^{-1}$$

 **$\mathbf{S}$  estimate:**
**for**  $i$  from 1 to  $R$  **do**

$$S_{j^*i} = 1 \iff j^* = \operatorname{argmax}_j \frac{\langle \mathbf{B}_i | \mathbf{D}_j \rangle}{\|\mathbf{D}_j\|}$$

**end for**
 **$\lambda$  update**
**if**  $\|\mathbf{B} - \mathbf{D} \mathbf{S}\|_F^2 > 0.01 \|\mathbf{B}\|_F^2$  **then**

$$\lambda = p\lambda$$

**end if**
**OUTPUT:** Estimated scores  $\mathbf{S}$  and factor  $\mathbf{B}$ .

---

fully flexible, therefore called Flex-MPALS. In S-MPALS, until  $\|\mathbf{B} - \mathbf{D} \mathbf{S}\|_F^2 \geq 0.01 \|\mathbf{B}\|_F^2$  is reached, the coupling strength  $\lambda$  increases by a multiplicative constant  $p$ . A relatively aggressive choice for the increase  $p = 1.1$  for third-order tensors. For matrices, even a higher value of  $p = 1.5$  gave good results and allowed faster convergence. In Flex-MPALS, the user chooses a parameter  $\beta$  so that when  $\|\mathbf{B} - \mathbf{D} \mathbf{S}\|_F^2 \geq \beta \|\mathbf{B}\|_F^2$ , while  $\lambda$  is increased in a similar way.

### B. Continuous approaches

As explained above, the original optimization problem underlying the DCPD model is combinatorial because of the  $\ell_0$  pseudo-norm and the fact that  $\mathbf{S}$  is binary. To develop continuous algorithms such as gradient descent, a first step is to derive a continuous relaxation for the DCPD formulation. The first relaxations that comes to mind when working on  $\ell_{col,0}$  pseudo-norm is a mixed norm  $\ell_{2,1}$  which encourages sparsity column-wise. However in the DCPD model,  $\mathbf{S}$  can only have one non-zero coefficient per column, which makes a  $\ell_{2,1}$  based optimization difficult to tune. Rather than mixed norm, we choose to use the  $\ell_1$  norm as a sparsity enhancing penalization on all the entries of  $\mathbf{S}$ , but under the constraint that the  $\ell_2$  norm of the columns of  $\mathbf{S}$  are set to one (which is not convex). Indeed, if  $\mathcal{B}(1, \ell_2)$  is the unit ball of the  $\ell_2$  norm, the solutions to

$$\operatorname{argmin}_{x \in \mathcal{B}(1, \ell_2)} \|x\|_1 \quad (15)$$

are exactly located on the coordinate axes, which is also the constraint imposed on the columns of  $\mathbf{S}$ . In light of this remark, solely imposing a  $\ell_1$  penalization on coefficients of  $\mathbf{S}$ , that could be easily done with for instance ADMM, may not be sufficient to ensure the high level of column sparsity that is sought.

We chose to use the fast gradient from Nesterov [55] which can be understood as a proximal gradient descent with averaged steps. Although our problem is not convex, this method can still be applied and has been shown to work well in the non-convex case as well [56]. Fast gradient is used to solve the following subproblem in  $\mathbf{S}$

$$\begin{cases} \min_{\mathbf{S}} \gamma(\mathbf{S}) = \frac{1}{2} \|\mathcal{T} - (\mathbf{A} \otimes \mathbf{D} \mathbf{S} \otimes \mathbf{C}) \mathcal{I}_R\|_F^2 + \delta \|\mathbf{S}\|_1 \\ \text{such that } \|\mathbf{s}_i\|_2^2 = 1 \quad \forall i \leq R, \quad \mathbf{S} \geq 0, \end{cases} \quad (16)$$

for a given parameter  $\delta$  fixed by the user, and where  $\mathbf{s}_i$  is the  $i^{th}$  column of  $\mathbf{S}$ . The non-negativity constraint on  $\mathbf{S}$  makes  $\gamma$  differentiable with respect to  $\mathbf{S}$ :

$$\frac{\partial \gamma}{\partial \mathbf{S}} = \mathbf{D}^\top \mathbf{D} \mathbf{S} (\mathbf{A}^\top \mathbf{A} \boxplus \mathbf{C}^\top \mathbf{C}) - \mathbf{D}^\top \mathcal{T}_2 (\mathbf{A} \odot \mathbf{C}) + \delta \mathbf{1}_{d \times R} \quad (17)$$

and the gradients of (16) with respect to  $\mathbf{A}$  and  $\mathbf{C}$  can be found for instance in [57] (we set  $\mathbf{B} = \mathbf{D}\mathbf{S}$ ) if an all-at-once optimization is sought. The normalization constraint on the columns of  $\mathbf{S}$  is imposed by normalization of  $\mathbf{S}$  at each iteration.

The global optimization procedure we suggest is therefore a mix of ALS and fast gradient, denoted ALS-FG and summarized in Algorithm 3. An important remark is that this algorithm is to be used inside

---

**Algorithm 3** Fast Gradient for estimating  $\mathbf{S}$

---

**INPUT:** array  $\mathcal{T}$ , factors  $\mathbf{A}$ ,  $\mathbf{C}$ , initial scores  $\mathbf{S}$ , dictionary  $\mathbf{D}$ , regularization parameter  $\delta > 0$ , fast gradient parameter  $\alpha \in ]0, 1]$ .

step size computation:

$\epsilon_S =$  product of squared largest eigenvalues  
of  $\mathbf{D}^T \mathbf{D}$  and  $(\mathbf{A}^T \mathbf{A} \boxplus \mathbf{C}^T \mathbf{C})$

**while** convergence criterion is not met **do**

gradient computation:

$$\mathbf{g}_S = \mathbf{D}^T \mathbf{D} \mathbf{S} (\mathbf{A}^T \mathbf{A} \boxplus \mathbf{C}^T \mathbf{C}) - \mathbf{D}^T \mathbf{T}_2 (\mathbf{A} \odot \mathbf{C}) + \delta \mathbf{1}_{d \times R}$$

gradient descent correction:

$$\mathbf{S}_{old} = \mathbf{S}$$

$$\mathbf{S} = \max(0, \mathbf{S} - \frac{1}{\epsilon_S} \mathbf{g}_S)$$

normalize columns of  $\mathbf{S}$  using the  $\ell_2$  norm

$$\alpha_{old} = \alpha$$

$$\alpha = \frac{1}{2}(-\alpha_{old}^2 + \sqrt{\alpha_{old}^4 + 4\alpha_{old}})$$

$$\beta = \frac{\alpha_{old}(1-\alpha_{old})}{(\alpha_{old}^2 + \alpha)}$$

update:

$$\mathbf{S} = \mathbf{S} + \beta(\mathbf{S} - \mathbf{S}_{old})$$

$$\mathbf{B} = \mathbf{D} \mathbf{S}$$

**end while**

**OUTPUT:** Estimated factor and scores  $\mathbf{B}$  and  $\mathbf{S}$ .

---

an outer loop consisting of least squares problems. To ensure a smooth transition from the unconstrained problem to enforcing the dictionary, the penalty coefficient  $\delta$  is linearly increased up from zero to a maximum value at the last outer iteration specified by the user. We observed that using the fast gradient improves convergence speed with respect to a simple coordinate descent, but other methods can also be implemented to solve the subproblem. Again, a flexible version of ALS-FG tackling optimization problem (9) can be derived from Algorithm 3 by adapting the gradient of  $\gamma$  with respect to  $\mathbf{S}$  and using  $\mathbf{B}$  as a variable.

*a) Stopping criterion:* Choosing the stopping criterion is application-dependent, and we let this parameter be tuned by interested users. Nevertheless, a baseline that we used in simulations below is to compute the residual error  $E_i$  at iteration  $i$  of the global optimization procedure, every few iterations, and to check whether the error is still decreasing enough by computing  $\frac{|E_i - E_{i-1}|}{E_i}$ . We set the number of iterations of the fast gradient inside ALS-FG to 10.

*b) Normalization:* An inherent ambiguity of the CPD model which is also present in the DCPD model is the scaling ambiguity, that is, the norm of columns of the factors is not determined solely by the model. It is possible to fix the norm of factors in both MPALS and ALS-FG by normalizing the columns of factor  $\mathbf{A}$  after each update of this factor, but this is not mandatory.

*c) Initialization:* Because the global optimization problems underlying the DCPD model and its flexible counterparts are non-convex and the proposed optimization algorithms work locally, it is *a priori* crucial to use a good initial point. If any information is available on the factors like non-negativity, it should be used in the initialization procedure. However, in the general case where no such information is available, one possible strategy is to compute an unconstrained CPD and use the obtained factors as an initialization for MPALS and ALS-FG. In the experiments below, we observed that indeed MPALS is

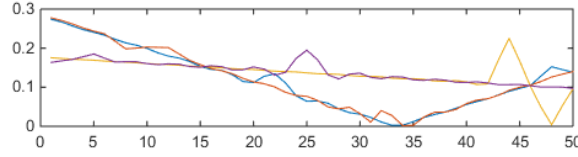


Fig. 1. Four dictionary atoms belonging to two different classes.

highly sensitive to initialization. In particular, using a single random initialization leads in many cases to poor results for third-order tensors. Note however that it worked well for hyperspectral images.

*d) Computing the Dictionary PARAFAC2:* Even though the PARAFAC2 flexible version of DCPD seems complicated to actually compute, it has been proven [28] that its computation only requires computing singular value decompositions (SVD) and the CPD. In our case we also simply need SVD, an algorithm to compute CPD and a DCPD algorithm like MPALS. SVD is used to estimate matrices  $\mathbf{P}_m$  for all  $m \leq M$  then each matrix  $\mathbf{M}_m$  is preprocessed by multiplying it with  $\mathbf{P}_m^T$ . Finally by stacking the preprocessed slices  $\mathbf{M}_m \mathbf{P}_m^T$  along the third mode, a tensor following a DCPD model is obtained, which can be decomposed using an alternating least squares algorithm featuring MPALS. The whole procedure can be repeated until convergence, and we call the obtained algorithm P2-MPALS<sup>5</sup>.

## VI. EXPERIMENTS ON SIMULATED AND HYPERSPECTRAL DATA

### A. Simulated data experiments

*1) Methodology:* A critical question yet unanswered is whether using a dictionary within the tensor factorization model improves on several criterion with respect to only projecting the result of an unconstrained factorization on the set of atoms. In particular, in this simulation subsection, we study the impact of dictionary CPD with respect to unconstrained CPD on the mean squared reconstruction error of the factor matrices, and with respect to identification error, that is, the percentage of columns of  $\mathbf{B}$  correctly matched to atoms of  $\mathbf{D}$ . Because information is added about factor  $\mathbf{B}$  in the model, the estimation error on this factor is expected to be smaller by using DCPD, in particular if the simulations are made challenging in terms of signal to noise ratio (SNR) or conditioning of the tensor (amount of colinearity between the columns of factors). However if the SNR is high, the tensor is well-conditioned and the rank is known, unconstrained CPD may provide relatively low identification error because of the uniqueness of the solution.

For this reason, we choose difficult scenarios to compare DCPD to unconstrained CPD with outputs identified using a projection on the set of atoms. The dictionary used is constituted of atoms as follows:

$$\mathbf{d}_i^k = |a_k \mathbf{u} + b_k + \nu_k^i \text{sinc}(\frac{\pi}{6} \mathbf{u} - e_k^i) + \mu_k^i (\text{tri}(\mathbf{u} - f_k^i + 2) - \text{tri}(\mathbf{u} - f_k^i - 2))| \quad (18)$$

where  $\mathbf{u} = [1..L]$ ,  $a_k, b_k, \nu_k^i, \mu_k^i, e_k^i, f_k^i$  are realizations of uniformly distributed random variables on respectively  $[-1/L, 1/L]$ ,  $[-1, 1]$ ,  $[-1/4, 1/4]$ ,  $[-1/4, 1/4]$ ,  $1..L$ ,  $1..L$  and tri is a triangular pulse function, of support  $[-2, 2]$ . Atoms are then normalized with the  $\ell_2$  norm. Index  $k \in \{1, \dots, c\}$  is a class index,  $c$  being the number of classes, and  $i \in \{1, \dots, \frac{d}{c}\}$  indexes individual atoms in each class. This means that atoms of the dictionary are a sum of a linear baseline common to groups of atoms and of two individual features. This design is meant to bluntly echo spectral signatures of materials in hyperspectral imaging, where only a small number of features can be used to differentiate materials from the same family; see Figure 1 for an example.

The data are generated as follows: factors  $\mathbf{A}$  and  $\mathbf{C}$  are drawn according to a unitary centered Gaussian distribution, and their columns are then normalized with the  $\ell_2$  norm, which makes these factors well-conditioned matrices. The matrix  $\mathbf{S}$  is chosen randomly so that several atoms of each class can be contained in the columns of  $\mathbf{B}$ . White Gaussian noise of fixed variance  $\sigma$  is added to the data tensor.

<sup>5</sup>The code for P2-MPALS is available online although it is not discussed in the simulation section.

	Identif. rate (%)	Factor relative error (%)			average time (s)
		<b>A</b>	<b>B</b>	<b>C</b>	
NWAY	—	23.27	0.99	24.82	4.52
NWAY-Proj.	74.4	23.27	0.42	24.82	—
MPALS	<b>78.6</b>	17.27	<b>0.31</b>	18.71	0.85
rand-MPALS	24.9	80.65	5.06	80.40	4.31
SMPALS	76.1	18.30	<b>0.31</b>	19.36	1.07
Flex-MPALS	76.8	<b>15.66</b>	0.62	<b>17.19</b>	1.13
ALS-FG	74.8	18.94	0.36	18.73	29.1

TABLE III  
RESULTS OF THE UNCONSTRAINED DATA EXPERIMENT. BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Simulations are conducted for unconstrained data and non-negative data. In the non-negative case, the absolute value of factors is taken for generating the data tensor. Non-negativity makes the low-rank approximation problem well-posed and is a constraint likely to be met in applications.

Given the structure of the dictionary, the correlation among atoms of the same class can be quite high. The dictionary is constituted of a large number of atoms ( $d = 1000$ ) subdivided in  $c = 50$  groups. The dimensions of the tensors are  $20 \times 50 \times 20$ , yielding an inter-group correlation of atoms that reaches 0.999 at most. We choose a noise variance of  $\sigma = 0.01$  which leads to an average SNR of about 6dB (about 13.3dB in the non-negative case)<sup>6</sup>, and set the rank to  $R = 10$ . To make the problem even more difficult, we chose to overestimate the rank by setting the number of estimated components to 12. Atoms involved in the actual decomposition (the true  $S$ ) are chosen randomly without repetition.

The CPD algorithms used as a baseline for comparison and for initialization, in both unconstrained and non-negative cases, is the NWAY toolbox [58], while we used the accelerated Hierarchical ALS (A-HALS) from [53] to compute factors  $A$  and  $C$  in the non-negative case. A-HALS runs similarly asymptotically to the fast-NNLS used in the NWAY toolbox, so that differences in performances should not be caused by using one non-negative least squares method or the other. For both unconstrained and non-negative data, we compare projected results of the NWAY toolbox with MPALS, SMPALS, Flex-MPALS and ALS-FG. Coupling strength in Flex-MPALS is fixed in this experiment for reproducibility reasons, we choose  $\lambda = 0.04$ . The number of iterations is fixed to a maximum of 1000, except for the rand-MPALS where the maximum iteration number is 5000. Algorithm also stops if the stopping criterion mentioned in the previous section goes below  $10^{-8}$ .

Finally, in both cases we use the NWAY toolbox for initialization, except for the randomly initialized MPALS which is initialized with factors following the same distribution as the true factors.

2) *Results:* Tables III and report the mean relative reconstruction error  $\frac{\|A - \Pi(\hat{A})\|_F^2}{\|A\|_F^2}$  on each normalized estimated factor and the identification ratio that measures the percentage of well-matched columns between  $B$  and  $\hat{B}$ . Here  $\Pi$  is an optimal permutation and selection operator that matches  $R$  among  $R + 2$  columns of estimated factors with the columns of the true factors, computed using the Hungarian algorithm<sup>7</sup>. Note that for the NWAY toolbox, the reconstruction error on  $B$  is computed before the projection on the set of atoms, while NWAY-Proj refers to estimation error after projection and serves as a baseline method.

The DCPD model perfectly fits the simulated data, and as expected the reconstruction error decreases on  $B$  with respect to NWAY when using MPALS and its variants. The identification rate is also higher than the baseline with all methods, which is a direct but non-obvious consequence of the increased estimation performances. Note that the estimation error also decreases for the other two factors in the unconstrained case.

From the poor results of the randomized initialization for MPALS, it can be inferred that the proposed algorithms to compute DCPD are very sensitive to initialization, fact also supported by the next experiment.

<sup>6</sup>Because of the absolute value taken for the factors in the nonnegative case,  $\sigma = 0.01$  lead to a higher SNR.

<sup>7</sup>We use the code available at <http://www.mathworks.com/matlabcentral/fileexchange/20328-munkres-assignment-algorithm>.

	Identif. rate (%)	Factor relative error (%)			average time (s)
		<b>A</b>	<b>B</b>	<b>C</b>	
NWAY	—	12.02	2.68	12.03	3.52
NWAY-Proj.	60.2	12.02	0.42	12.03	—
MPALS	<b>65.5</b>	<b>10.73</b>	<b>0.25</b>	<b>10.76</b>	3.37
rand-MPALS	48.2	21.73	1.21	22.25	5.06
SMPALS	62.6	12.49	0.30	12.50	2.51
Flex-MPALS	61.8	12.42	0.50	13.26	4.30
ALS-FG	60.5	11.92	0.32	12.15	15.99

TABLE IV  
RESULTS OF THE NON-NEGATIVE DATA EXPERIMENT. BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Moreover, the smooth version of MPALS does not particularly improve estimation and identification performances. Finally, from this experiment we observe that computing the DCPD with a combinatorial greedy approach performs better than a continuous approach, since the performance of ALS-FG is quite poor, or at least that designing a continuous algorithm to compute DCPD requires particular and careful attention, and is a topic for further research. We believe the reason is that the greedy approaches, because of their projection steps, can escape the basin of attraction of a local minimum, which a (standard) continuous approach will not be able to do. In fact, we observed that, in most cases, once an entry is set to zero by ALS-FG, it remains zero in the course of the iterations, which is not the case for MPALS and its variants.

### B. Spectral Unmixing with the self-dictionary model

For this communication on dictionary based tensor factorization models, we choose to try out the different models and algorithms on a well-known sparse coding problem, namely spectral unmixing under the pure-pixel assumption. As explained in section II, the data itself can be used as the dictionary, and models described in this paper for matrices can be straightforwardly compared with state-of-the-art sparse coding approaches for spectral unmixing, namely the successive projection algorithm (SPA) [37], the successive nonnegative projection algorithm (SNPA) [10], the Hierarchical Clustering algorithm (H2NMF) [59] and FGNSR [40]. Below, the spectra contained in the data are used as atoms, so that the dictionary has row dimension equals to the number of spectral bands ( $\approx 150$ ), and number of atoms equals to the number of pixels in the hyperspectral images (HSI) ( $\approx 10^5$ ).

The three data sets that will be used to compare DCPD with these methods are Urban, Terrain and San Diego Airport HSIs. These three HSIs satisfy approximately the pure pixel assumption. The dimensions of the HSI after vectorization of the pixel dimensions are respectively  $307^2 \times 162$ ,  $(500 \times 307) \times 166$  and  $400^2 \times 158$ . We chose to decompose the two data sets with respective ranks  $R_{urban} = \{6, 8\}$ ,  $R_{terrain} = \{5, 6\}$  and  $R_{SDA} = \{8, 10\}$  according to what has been done previously in the literature; see [40] and the references therein. The maximal number of iterations of the DCPD algorithms is set to 50, except the Flex-MPALS which performs a maximum of 250 iterations that is almost always reached. For Flex-MPALS, the initial  $\delta$  was fixed to 100.

Also because it does not perform well, results for the continuous fast gradient algorithm are not presented below. For the flexible MPALS, parameter  $\delta$  is increased every twenty iterations if the estimated factor is further than 10% to the dictionary in relative Frobenius norm.

We initialize the MPALS algorithm and its variants with each state-of-the-art algorithm with 10 additional A-HALS steps, for instance MPALS and SMPALS initialized with SPA are respectively denoted as d-SPA and ds-SPA. Results are presented in Table V.

In all cases, using either SMPALS or MPALS improves on the initial values for identified spectra. As shown already in the previous experiment, initialization plays an important role in the final reconstruction error, but at least using the DCPD model always refines the solutions, even when it is initially low like



	Urban HSI				Terrain HSI				SDA HSI			
	$R = 6$		$R = 8$		$R = 5$		$R = 6$		$R = 8$		$R = 10$	
	t(s)	Rel.err.	t(s)	Rel.err.	t(s)	Rel.err.	t(s)	Rel.err.	t(s)	Rel.err.	t(s)	Rel.err.
SPA	10	9.58	12	9.45	13	5.89	14	4.81	9	12.62	10	7.01
d-SPA	36	4.57	55	4.47	56	3.50	71	4.15	110	3.96	118	3.97
ds-SPA	36	4.67	54	4.62	52	3.37	72	3.81	109	4.08	118	3.91
Flex-SPA	107	3.20	146	2.79	167	2.49	202	2.32	237	2.54	312	2.17
VCA	5	13.07	5	18	6	18.61	6	6.9	8	15.82	29	6.28
d-VCA	31	4.73	42	5.35	37	3.29	49	3.13	80	4.78	90	6.25
ds-VCA	27	4.66	51	4.99	35	3.37	49	3.13	109	4.19	92	6.25
Flex-VCA	108	3.46	154	3.17	161	2.54	196	3.20	220	2.27	340	1.86
SNPA	20	9.63	31	5.64	27	5.76	36	4.60	58	12.84	81	7.67
d-SNPA	25	5.02	48	4.43	58	3.78	61	3.92	107	<b>3.58</b>	151	4.20
ds-SNPA	27	4.91	51	<b>3.97</b>	62	3.88	61	3.70	110	3.75	147	4.45
Flex-SNPA	110	3.11	133	2.76	164	2.46	196	2.22	282	2.51	340	2.17
H2NMF	21	5.81	27	5.47	31	5.09	38	4.85	42	4.75	45	4.28
d-H2NMF	31	<b>4.05</b>	48	4.23	67	3.49	72	3.16	157	4.21	210	<b>3.28</b>
ds-H2NMF	31	<b>4.05</b>	48	4.24	65	3.49	68	3.30	154	4.13	218	3.36
Flex-H2NMF	105	3.00	132	2.66	163	2.48	196	2.28	298	2.77	361	2.09
FGNRS-100	10	5.58	10	4.62	14	3.34	11	3.21	20	3.73	22	3.40
d-FGNSR-100	32	4.47	48	4.23	56	<b>3.01</b>	73	<b>2.76</b>	110	3.63	186	3.55
ds-FGNSR-100	31	4.47	49	4.04	55	<b>3.01</b>	73	2.84	110	3.63	185	<b>3.28</b>
Flex-FGNSR-100	111	3.19	131	2.46	163	2.43	196	2.21	237	2.43	302	1.95

TABLE V

RECONSTRUCTION ERROR (%) FOR THE URBAN, TERRAIN AND SDA AIRPORT HSI. BEST RESULTS ARE HIGHLIGHTED IN BOLD. COMPUTATION TIME INCLUDE 500 ITERATIONS OF NONNEGATIVE LEAST SQUARES UPDATE FOR THE ABUNDANCES AFTER EACH METHOD.

for FGNSR-100. Moreover, the DCPD algorithms are not excessively costly with respect to other state-of-the-art methods. There is no significant difference between MPALS and its smooth counterpart. Figure 2 shows the estimated abundances and spectra with the H2NMF-MPALS algorithm for both Urban and Terrain, and materials can be identified for each component by the user (no ground truth is available). Finally, Figure 3 shows the reconstruction error map on Urban and Terrain HSIs for the H2NMF-MPALS. Clearly the remaining error is not distributed as an i.i.d. Gaussian noise, which means the Frobenius norm used as the distance metric is not adapted. Also, most of the remaining error comes from rooftops and roads in the Urban HSI, a zone probably corrupted by large spectral variability.

Because Flex-MPALS does not require exactly that the pure pixel assumption is verified, using a reconstruction error criteria, Flex-MPALS outperforms the other methods, but that does not mean that obtained abundances and endmembers are better. On the other hand, the flexible model better tackles spectral variability since it can modify to some extent the spectra extracted from the pure pixels.<sup>8</sup> For instance, in Figure 5 showing the residual signal after estimating the abundance maps and the spectral signatures using Flex-MPALS, it can be seen that the portion of the URBAN scene that is a rooftop (top right) is better estimated than in the non-flexible approach. Moreover this part of the scene is known to be prone to spectral variability because of high reflectance. The abundance maps estimated using the flexible algorithm and shown in Figure 4 also have more discriminative features, which means that hopefully the abundance maps are more accurate with respect to the unknown ground truth.

## VII. CONCLUSIONS

To impose identification of estimated sources in tensor canonical polyadic decomposition according to a known dictionary, we introduced in this paper the DCPD model along with some flexible variants.

<sup>8</sup>The code for matrix Flex-MPALS is also online.

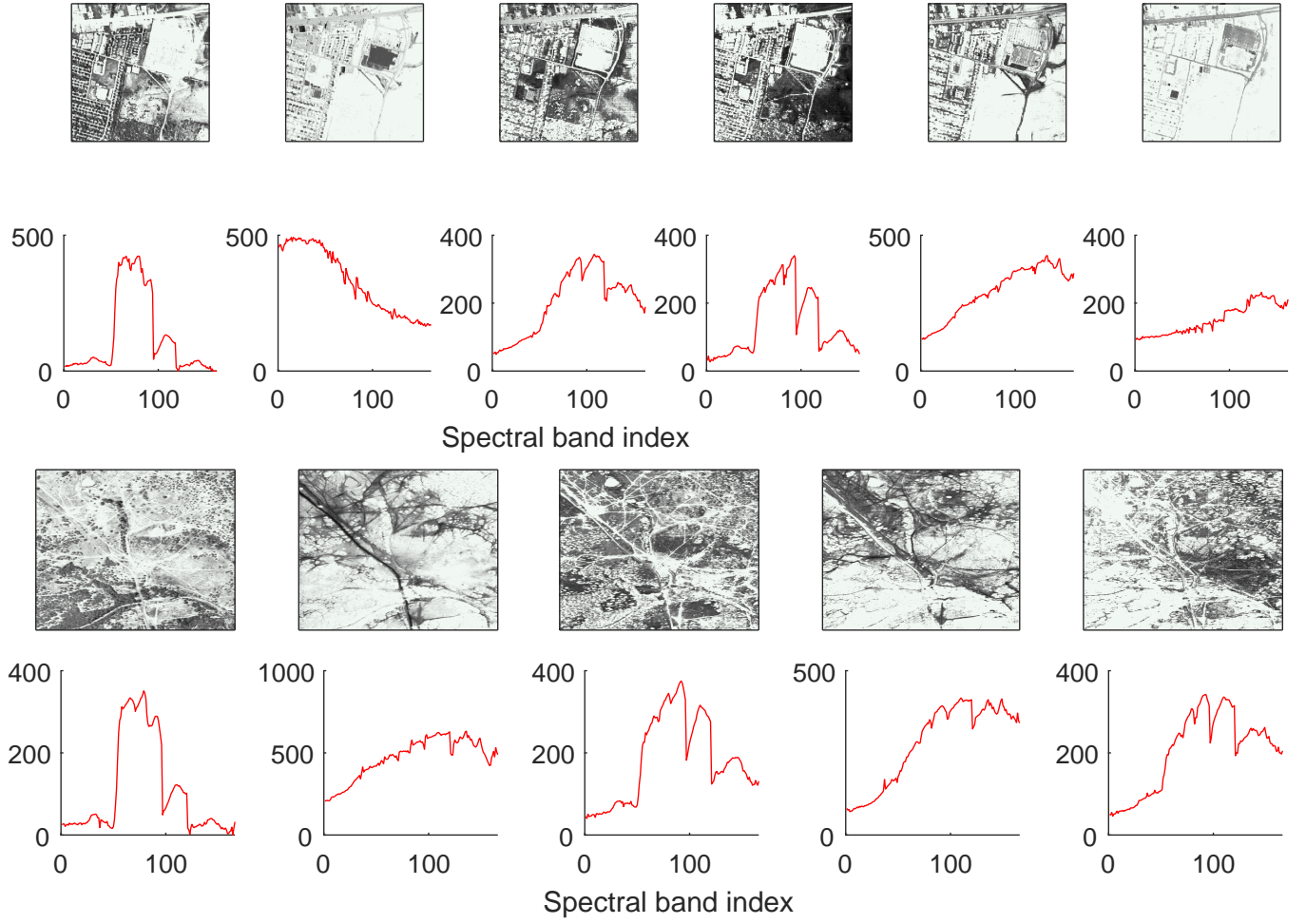


Fig. 2. Estimated spectra on the Urban HSI with d-H2NMF (top) and on the Terrain HSI with d-H2NMF (bottom).

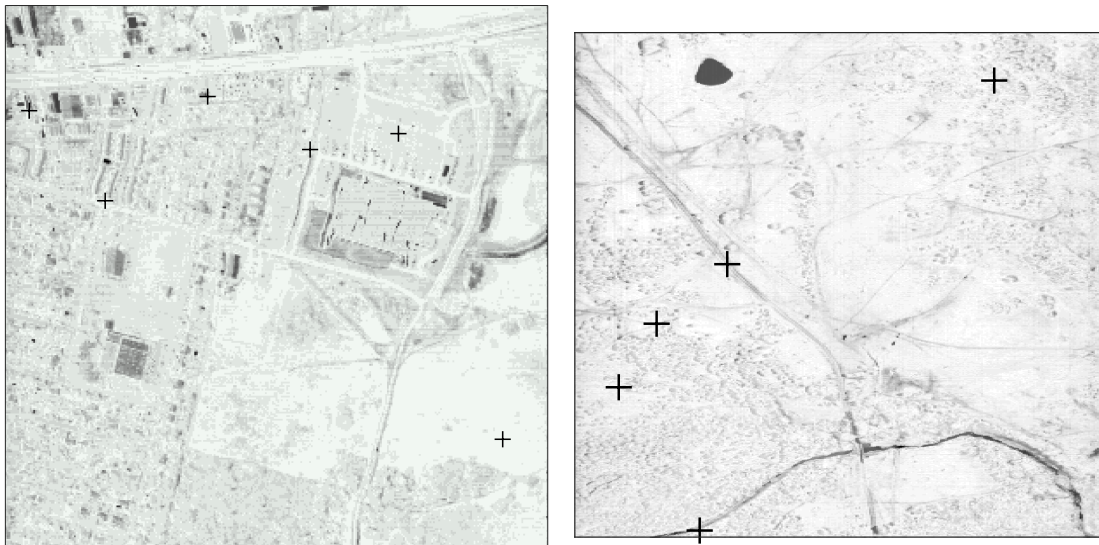


Fig. 3. Residual error maps on the Urban (left) and Terrain (right) HSIs for d-H2NMF. The black crosses mark the selected pure pixels.

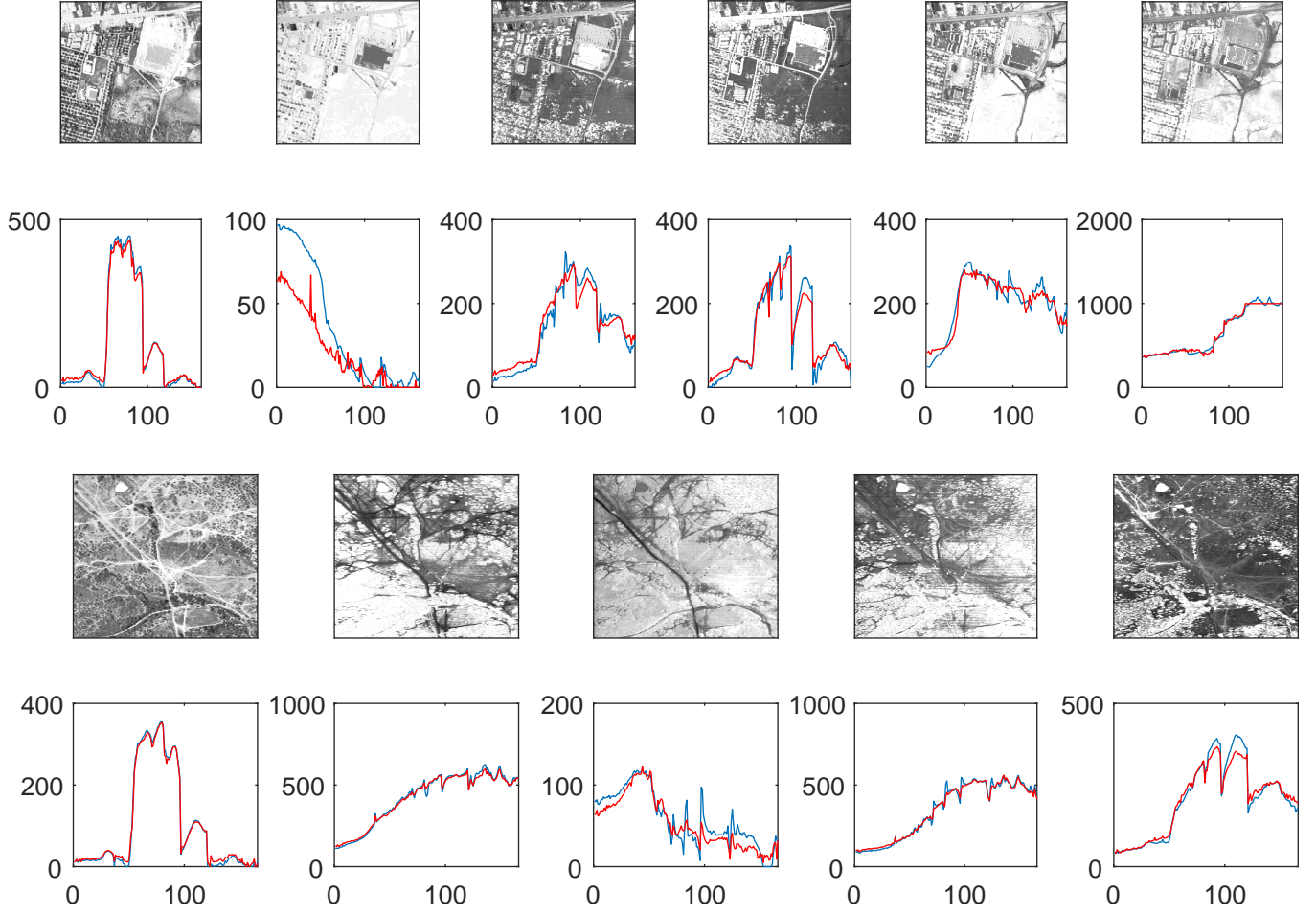


Fig. 4. Estimated spectra on the Urban HSI with Flex-H2NMF (top) and on the Terrain HSI with Flex-H2NMF (bottom). In red is the selected atom in the dictionary, while in blue is the estimated spectral signature.

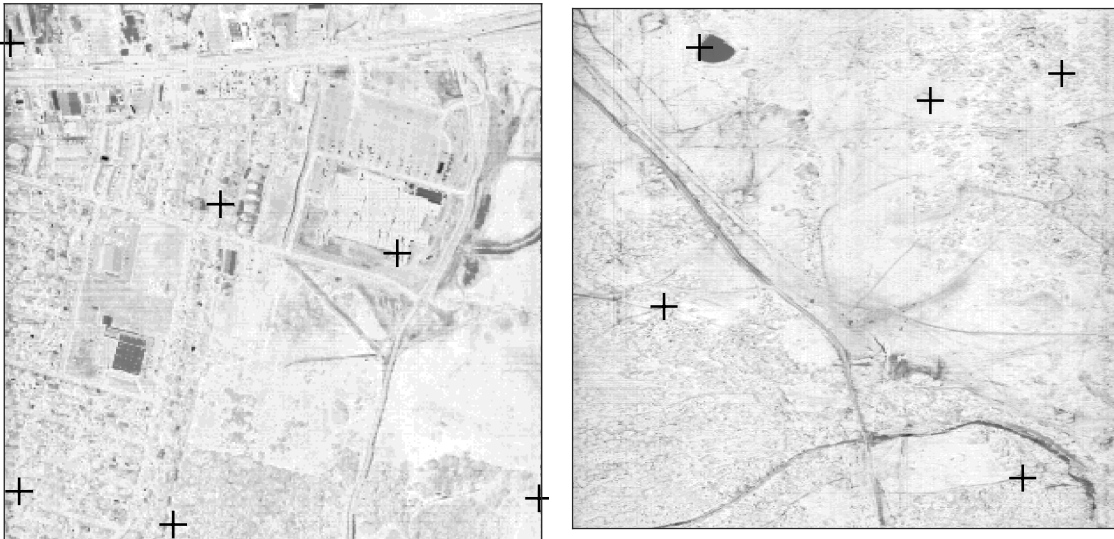


Fig. 5. Residual error maps on the Urban (left) and Terrain (right) HSIs for Flex-H2NMF. The black crosses mark the selected pure pixels.

Identifiability of the DCPD model parameters was discussed in both the matrix and the higher-order tensor cases. We proposed both greedy and continuous algorithms for DCPD and compared them on synthetic data sets and hyperspectral images. We observed that (i) the greedy algorithms provide in most cases better results, and that (ii) our DCPD formulation improves results with respect to standard CPD on the synthetic data sets, and with respect to spectral unmixing approaches based either on convex relaxation or geometric methods.

An interesting continuation of this work would be to compute the Cramér Rao bounds of the parameters in the DCPD to formally support what was presented in the simulation section. Moreover, in most application cases the dictionary is either unknown, or describes only a subset of the components. Thus a partial DCPD model should be investigated, as well as a dictionary learning scheme for tensors following the DCPD model.

### ACKNOWLEDGMENT

The authors would like to thank Pierre Comon, Rodrigo Cabral Farias and Miguel A. Veganzones for early fruitful discussions on this work.

### REFERENCES

- [1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.
- [2] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Trans. Sig. Proc.*, vol. 48, no. 8, pp. 2377–2388, Aug. 2000.
- [3] R. Bro, "Multi-way Analysis in the Food Industry: Models, Algorithms, and Applications," Ph.D. dissertation, University of Amsterdam, The Netherlands, 1998.
- [4] M. A. Veganzones, J. E. Cohen, R. Cabral Farias, J. Chanussot, and P. Comon, "Nonnegative tensor CP decomposition of hyperspectral data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, pp. 2577–2588, 2016.
- [5] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *Signal Processing Magazine, IEEE*, vol. 32, no. 2, pp. 145–163, 2015.
- [6] P. M. Kroonenberg, *Three-mode principal component analysis: Theory and applications*. DSWO press, 1983, vol. 2.
- [7] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, p. 16, 2016.
- [8] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.
- [9] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1600–1607.
- [10] N. Gillis, "Successive nonnegative projection algorithm for robust nonnegative blind source separation," *SIAM J. Imaging Sci.*, vol. 7, no. 2, pp. 1420–1450, 2014.
- [11] J. Nascimento and J. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2005.
- [12] A. Kumar, V. Sindhwani, and P. Kambadur, "Fast conical hull algorithms for near-separable non-negative matrix factorization," in *Int. Conf. on Machine Learning (ICML '13)*, 2013, vol. 28, no. 1, pp. 231–239.
- [13] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithms for separable nonnegative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 698–714, 2014.
- [14] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, ser. Series in Computational Mathematics. Berlin, Heidelberg: Springer, 2012.
- [15] J. E. Cohen, "About notations in multiway array processing," *arXiv preprint arXiv:1511.01306*, 2015.
- [16] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977.
- [17] I. Domanov and L. De Lathauwer, "On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part II: Uniqueness of the overall decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 876–903, 2013.
- [18] J. M. Landsberg and G. Ottaviani, "Equations for secant varieties of veronese and other varieties," *Annali di Matematica Pura ed Applicata*, vol. 192, no. 4, pp. 569–606, 2013.
- [19] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on circuits and systems*, vol. 25, no. 9, pp. 772–781, 1978.
- [20] C. R. Rao, *Linear Statistical Inference and its Applications*, ser. Probability and Statistics. Wiley, 1965.
- [21] B. Valeur and M. N. Berberan-Santos, *Molecular fluorescence: principles and applications*. John Wiley & Sons, 2012.
- [22] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [23] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, 2012.

- [24] A. Zare and K. Ho, "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 95–104, 2014.
- [25] R. Cabral-Farias, J. E. Cohen, and P. Comon, "Exploring multimodal data fusion through joint decompositions with flexible couplings," *IEEE Trans. Sig. Proc.*, vol. 64, no. 18, pp. 4830–4844, 2016.
- [26] Y. Guo and M. Berman, "A comparison between subset selection and  $l_1$  regularisation with an application in spectroscopy," *Chemometrics and Intelligent Laboratory Systems*, vol. 118, pp. 127–138, 2012.
- [27] R. A. Harshman, "PARAFAC2: Mathematical and technical notes," *UCLA working papers in phonetics*, vol. 22, no. 3044, p. 122215, 1972.
- [28] H. A. Kiers, J. M. Ten Berge, and R. Bro, "PARAFAC2-Part I. A direct fitting algorithm for the PARAFAC2 model," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 275–294, 1999.
- [29] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *In Advances in Neural Information Processing 16*, 2003.
- [30] S. Arora, R. Ge, R. Kannan, and A. Moitra, "Computing a nonnegative matrix factorization—provably," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012, pp. 145–162.
- [31] A. M. S. Ang, Y. S. Hung, and Z. Zhang, "A non-negative tensor factorization approach to feature extraction for image analysis," in *Digital Signal Processing (DSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 168–178.
- [32] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 341–354, 2014.
- [33] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, "A convex model for nonnegative matrix factorization and dimensionality reduction on physical space," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3239–3252, 2012.
- [34] V. Bittorf, B. Recht, E. Ré, and J. Tropp, "Factoring nonnegative matrices with linear programs," in *Advances in Neural Information Processing Systems (NIPS '12)*, 2012, pp. 1223–1231.
- [35] N. Gillis and R. Luce, "Robust near-separable nonnegative matrix factorization using linear optimization," *J. Mach. Learn. Res.*, vol. 15, pp. 1249–1280, 2014.
- [36] S. Sahnoun, E.-H. Djermoune, D. Brie, and P. Comon, "A simultaneous sparse approximation method for multidimensional harmonic retrieval," *Signal Processing*, vol. 131, pp. 36–48, 2017.
- [37] U. Araújo, B. Saldanha, R. Galvão, T. Yoneyama, H. Chame, and V. Visani, "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis," *Chemometr. Intell. Lab.*, vol. 57, no. 2, pp. 65–73, 2001.
- [38] H. Ren and C.-I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1232–1249, 2003.
- [39] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4177–4193, 2011.
- [40] N. Gillis and R. Luce, "A fast gradient method for nonnegative sparse regression with self dictionary," *arXiv:1610.01349*, 2016.
- [41] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [42] M. E. Timmerman and H. A. Kiers, "Three-way component analysis with smoothness constraints," *Computational statistics & data analysis*, vol. 40, no. 3, pp. 447–470, 2002.
- [43] I. Domanov and L. De Lathauwer, "Generic uniqueness of a structured matrix factorization and applications in blind source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 701–711, 2016.
- [44] D. A. Roberts, M. Gardner, R. Church, S. Ustin, G. Scheer, and R. Green, "Mapping chaparral in the santa monica mountains using multiple endmember spectral mixture models," *Remote Sensing of Environment*, vol. 65, no. 3, pp. 267–279, 1998.
- [45] J.-P. Combe, S. Le Mouelic, C. Sotin, A. Gendrin, J. Mustard, L. Le Deit, P. Launeau, J.-P. Bibring, B. Gondet, Y. Langevin *et al.*, "Analysis of omega/mars express data hyperspectral data using a multiple-endmember linear spectral unmixing model (melsum): Methodology and first results," *Planetary and Space Science*, vol. 56, no. 7, pp. 951–975, 2008.
- [46] C. Song, "Spectral mixture analysis for subpixel vegetation fractions in the urban environment: How to incorporate endmember variability?" *Remote Sensing of Environment*, vol. 95, no. 2, pp. 248–263, 2005.
- [47] G. P. Asner, M. M. Bustamante, and A. R. Townsend, "Scale dependence of biophysical structure in deforested areas bordering the tapajos national forest, central amazon," *Remote Sensing of Environment*, vol. 87, no. 4, pp. 507–520, 2003.
- [48] I. Domanov, "Study of canonical polyadic decomposition of higher-order tensors," Ph.D. dissertation, KU Leuven, 2013.
- [49] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.
- [50] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [51] —, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE transactions on information theory*, vol. 52, no. 3, pp. 1030–1051, 2006.
- [52] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of chemometrics*, vol. 11, no. 5, pp. 393–401, 1997.
- [53] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization," *Neural computation*, vol. 24, no. 4, pp. 1085–1105, 2012.
- [54] A. Cichocki and S.-i. Amari, *Adaptive blind signal and image processing: learning algorithms and applications*. John Wiley & Sons, 2002, vol. 1.
- [55] Y. Nesterov, "A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ ," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.
- [56] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," *Mathematical Programming*, vol. 156, no. 1-2, pp. 59–99, 2016.

- [57] J.-P. Royer, N. Thirion-Moreau, and P. Comon, "Computing the polyadic decomposition of nonnegative third order tensors," *Signal Processing*, vol. 91, no. 9, pp. 2159–2171, 2011.
- [58] C. A. Andersson and R. Bro, "The n-way toolbox for matlab," *Chemometrics and intelligent laboratory systems*, vol. 52, no. 1, pp. 1–4, 2000.
- [59] N. Gillis, D. Kuang, and H. Park, "Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sensing*, vol. 53, no. 4, pp. 2066–2078, 2015.